[illegible]

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/889,416	07/17/2001	Christian Goire	T2146-907404	3797

181 7590 12/01/2004

MILES & STOCKBRIDGE PC
1751 PINNACLE DRIVE
SUITE 500
MCLEAN, VA 22102-3833

EXAMINER	
SHRADER, LAWRENCE J	
ART UNIT	**PAPER NUMBER**
2124	

DATE MAILED: 12/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/889,416	GOIRE ET AL.	
	Examiner	Art Unit	
	Lawrence Shrader	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 7/17/2004; 8/13/2001.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-15 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-15 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 7/17/2001.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to the Applicant's application filed on 7/17/2004 and the preliminary amendments of 8/13/2001.

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted on 7/17/2001 is acknowledged. Accordingly, the information disclosure statement is being considered by the examiner.

Priority

3. Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d).

Specification

4. The abstract of the disclosure is objected to because the length exceeds 150 words. Correction is required. See MPEP § 608.01(b).

Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1, 5, 6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilkinson et al., U.S. Patent 6,308,317 (hereinafter referred to as Wilkinson) in view of Judge et al., U.S. Patent 6,430,570 (hereinafter referred to as Judge).

In regard to claim 1:

"assigning an identifier to each module (MID), and a reference number to classes of a module, so as to statically link a plurality of sets of packages to be stored in the same name space in the embedded system to effect conversion by the converter (92);"

Wilkinson discloses identification of classes, methods, attributes via the constant pool (column 8, line 62 to column 9, line 17).

"coding the reference to a method or an attribute in the linked pseudocode of a module in three bytes constituted by an indicator indicating the reference to a class internal (11) or external (IE) to the module, the number of the class (NCI), and either the number of the method (NM) or the number of the attribute (NA), and"

See Wilkinson column 9, lines 19 – 41; figures 3 and 4.

"loading one or more application program interface modules comprising system classes or service modules, each corresponding to an application, a reference (IE) to an external

class being systematically interpreted by the virtual machine as a reference to an application program interface module."

Wilkinson does not explicitly disclose reference to an external class as a reference to an API. However, Judge discloses a means for managing embedded systems wherein an API provides functionality for loading class files (column 6, lines 22 – 42). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of an API as a reference to load modules as taught by Judge because the API provides platform independence and efficiency as taught by Judge at column 1, lines 35 – 49.

In regard to claim 5, incorporating the rejection of claim 4:

"...characterized in that the classes being declared public, or in private packages, the attributes and methods being declared protected, in private packages or in private classes, the numbering of the classes is done in order of the public classes followed by the classes in private packages, the numbering of the attributes or methods is done by the converter (92) in order of the attributes or methods that are public, protected in private packages, and in private classes."

Wilkinson discloses the identification of attributes or methods done by the converter (column 9, lines 19 – 60).

In regard to claim 6, incorporating the rejection of claim 2:

"...further comprising containing system classes contained in several API modules that can be loaded separately, maintaining in the programmable nonvolatile memory two arrays representing modules and two corresponding MID/IMi association tables, one for the API modules and the other for the non-API modules, and loading the modules into one of the two arrays based on the nature of the module specified in its header, any external reference of a module of the module array being interpreted as a reference to the index of the API module."

Wilkinson does not explicitly disclose reference to an external class as a reference to an API. However, Judge discloses a means for managing embedded systems wherein an API provides functionality for loading class files in several API modules (e.g. see Figure 2). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of an API as a reference to load modules as taught by Judge because the API provides platform independence and efficiency as taught by Judge at column 1, lines 35 – 49.

7. Claims 2 – 4, 7; 8 – 13; 14 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilkinson et al., U.S. Patent 6,308,317 in view of Judge et al., U.S. Patent 6,430,570, and further in view of Hagy et al., U.S. Patent 6.363,436 (hereinafter referred to as Hagy).

In regard to claim 2, incorporating the rejection of claim 1:

> *"...characterized in that the loading of the modules into the embedded system comprises storing of at least one module array (69, 101 , 103) representing the modules, the number (IMi) between 0 and n associated by the linker with a module constituting the index of said module in the array, storing the association of the index in the array representing the identifier (MID) of said module in a table (70, 102, 104), said array and the table being in programmable nonvolatile memory, an external reference (IE) to an external module in the pseudocode interpreting as constituting an index for access to the module equivalent to the one in the module array."*

Wilkinson discloses identification of classes, methods, attributes via the constant pool, but neither Wilkinson nor Judge explicitly discloses an module index array. However, Hagy discloses an array of module pointers (column 4, lines 56 – 64). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of loader API as taught by Judge, and further modified by a module index array as taught by Hagy because the array provides an dependency list as described by Hagy at column 4, lines 60 – 63.

In regard to claim 3, incorporating the rejection of claim 2:

> *"...characterized in that the loading of the modules comprises the storage, for each module, of an array (TRC) representing its classes, comprising a reference to the index of its module and, for each class, an array representing attributes (TRA) and methods (TRMe)."*

Wilkinson discloses identification of classes, methods, attributes via the constant pool, but neither Wilkinson nor Judge explicitly discloses an module index array. However, Hagy discloses an array of module pointers (column 4, lines 56 – 64). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of loader API as taught by Judge, and further modified by a module index array as taught by Hagy because the array provides an dependency list as described by Hagy at column 4, lines 60 – 63.

In regard to claim 4, incorporating the rejection of claim 2:

> *"...characterized in that the modules are referenced in a single module array, interpreting system classes different from n by the virtual machine as a reference to said API module."*

Wilkinson discloses identification of classes, methods, attributes via the constant pool, but neither Wilkinson nor Judge explicitly discloses an module index array. However, Hagy discloses an array of module pointers (column 4, lines 56 – 64). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of loader API as taught by Judge, and further modified by a module index array as taught by Hagy because the array provides an dependency list as described by Hagy at column 4, lines 60 – 63.

In regard to claim 7, incorporating the rejection of claim 6:

> *"...characterized in that static linking of a module is performed such that the reference to a class external to a non-API module in the intermediate pseudocode is an index in an array of the header of the module, wherein each entry is an identifier (MID) of a referenced API module, the loading of said module into the target platform comprising the replacement of said reference with the number of the index of the API module obtained from the identifier (MID) in the MID/IMi association tables of the API modules."*

Wilkinson discloses identification of classes, methods, attributes via the constant pool, and uses static linking (column 9, lines 19 – 23), but neither Wilkinson nor Judge explicitly discloses an module index array. However, Hagy discloses an array of module pointers (column 4, lines 56 – 64). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of loader API as taught by Judge, and further modified by a module index array as taught by Hagy because the array provides an dependency list as described by Hagy at column 4, lines 60 – 63.

In regard to claim 8:

> *"An embedded system comprising a virtual machine and an API platform including application program interfaces, a fixed nonvolatile memory, a programmable nonvolatile memory, and a random access memory, the programmable nonvolatile memory comprising at least one API module comprising system classes and service modules, at least one array representing modules (TRM), in which the modules are indexed, and a table (70, 104) associating an index (IM) of a module in the representing array with an identifier (MID) of said module, each module comprising an array for representing*

classes (TRC), in which the classes are indexed and in which each class has a reference to the index (IM) of its module, each class comprising an array for representing attributes (TRA) and methods (TRMe), in which the attributes and methods are indexed, the reference to a method or an attribute being coded in at least three bytes corresponding to a reference to a class internal (II) or external (IE) to the module, a reference external to the module constituting the index of the API module in the module array, a class number (NCI) corresponding to the index of the class in the table representing the classes of the module, and a method (NM) or attribute (NA) number corresponding to the index or the method or the attribute in the array representing the methods or attributes of the class of the module."

Wilkinson discloses identification of classes, methods, attributes via the constant pool column 9, lines 9 – 41; figures 3 and 4), and Judge discloses a means for managing embedded systems wherein an API provides functionality for loading class files (column 6, lines 22 – 42), but neither Wilkinson nor Judge explicitly discloses an module index array. However, Hagy discloses an array of module pointers (column 4, lines 56 – 64). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of loader API as taught by Judge, and further modified by a module index array as taught by Hagy because the array provides an dependency list as described by Hagy at column 4, lines 60 – 63.

In regard to claim 9, incorporating the rejection of claim 8:

"...further comprising means for comparing the first byte of the three bytes encoding a reference to a method or an attribute to a given value n in order to decide whether it is an internal or an external class."

Wilkinson discloses identification of classes, methods, attributes via the constant pool column 9, lines 9 – 41; figures 3 and 4), but does not explicitly teach making a decision whether a class is internal or external. However, Official notice is taken that, for example, in a conventional Java system, references to class structures (internal and external) are resolved using indirect name lookup via a constant pool. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, attributes via the constant pool as taught by Wilkinson with the well known knowledge that references to class structures (internal and external) are resolved using indirect name lookup via a constant pool because the decision is inherent in the system.

In regard to claim 10, incorporating the rejection of claim 8:

"...wherein a main module comprises the main program of the system."

See Wilkinson column 8, lines 23 – 25.

In regard to claim 11, incorporating the rejection of claim 8:

"...characterized in that it the classes are indexed in order of the public classes followed by the classes in private packages, and the attributes and methods are indexed in the order of the attributes or methods that are public, protected, in private packages, and in private classes."

Wilkinson discloses the identification of attributes or methods done by the converter (column 9, lines 19 – 60).

In regard to claim 12, incorporating the rejection of claim 11:

"...characterized in that the programmable nonvolatile memory comprises several API modules comprising system classes, two arrays (101, 103) representing modules, one for the API modules and the other for the non-API modules and the main module, and two MID/IMi association tables (102, 104), each corresponding to an array representing modules."

Wilkinson does not explicitly disclose reference to an external class as a reference to an API. However, Judge discloses a means for managing embedded systems wherein an API provides functionality for loading class files in several API modules (e.g. see Figure 2). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of an API as a reference to load modules as taught by Judge because the API provides platform independence and efficiency as taught by Judge at column 1, lines 35 – 49.

In regard to claim 13, incorporating the rejection of claim 10:

> *"...further comprising an access manager class "Access Manager" of an API module (105) comprising a method for creating an instance of a service module, via the main module, said class having a protection that prohibits it from having more than one instance."*

Wilkinson does not explicitly disclose reference to an external class as a reference to an API. However, Judge discloses a means for managing embedded systems wherein an API provides functionality for loading class files by load class in the Application manager (column 4, line 39 to column 5, line 15; e.g., Figure 2). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of an API as a reference to load modules as taught by Judge because the API provides platform independence and efficiency as taught by Judge at column 1, lines 35 – 49.

In regard to claim 14:

> *"A method for executing an application of a multi-application embedded system having a runtime environment including a virtual machine comprising an intermediate pseudocode language interpreter and application programming interfaces (API), is characterized in that during the execution of the intermediate pseudocode of a service module, corresponding to an application, referenced in a module array, the reference to a method or an attribute in the pseudocode, coded in at least three bytes corresponding to a reference to a class internal (II) or external (IE) to the module, a class number (NCI) and a method (NM) or attribute (NA) number, a reference external to the module is*

> *interpreted by the virtual machine as a reference to the index of an API module in the array of the API module or modules."*

Wilkinson discloses identification of classes, methods, attributes via the constant pool column 9, lines 9 – 41; figures 3 and 4), and Judge discloses a means for managing embedded systems wherein an API provides functionality for loading class files (column 6, lines 22 – 42), but neither Wilkinson nor Judge explicitly discloses an module index array. However, Hagy discloses an array of module pointers (column 4, lines 56 – 64). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of loader API as taught by Judge, and further modified by a module index array as taught by Hagy because the array provides an dependency list as described by Hagy at column 4, lines 60 – 63.

In regard to claim 15, incorporating the rejection of claim 14:

> *"...characterized in that, upon reception of a request for execution of a service module having an identifier (MID), the runtime environment accesses the input class of a main module comprising the main program of the system, the main module installs an instance of a special class "Access Manager" of an API module that controls access to a service module and uses a method of this class for creating an instance of the input class of the requested service module, by means of a table associating the identifier with the index of the module in an array in which the module is referenced, the instance being returned by the method to the main program."*

Wilkinson does not explicitly disclose reference to an external class as a reference to an API. However, Judge discloses a means for managing embedded systems wherein an API provides functionality for loading class files by load class in the Application manager (column 4, line 39 to column 5, line 15; e.g., Figure 2). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the identification of classes, methods, and attributes as taught by Wilkinson with the use of an API as a reference to load modules as taught by Judge because the API provides platform independence and efficiency as taught by Judge at column 1, lines 35 – 49.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lawrence Shrader whose telephone number is (571) 272-3734. The examiner can normally be reached on M-F 08:00-16:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Lawrence Shrader
Examiner
Art Unit 2124

November 24, 2004

[illegible]
TODD INGBERG
PRIMARY EXAMINER